Stat Comput (2012) 22:795–808
DOI 10.1007/s11222-011-9231-6

Sequential Monte Carlo for rare event estimation

F. Cérou · P. Del Moral · T. Furon · A. Guyader

Received: 26 July 2010 / Accepted: 11 January 2011 / Published online: 5 April 2011

Abstract This paper discusses a novel strategy for simulating rare events and an associated Monte Carlo estimation of tail probabilities. Our method uses a system of interacting particles and exploits a Feynman-Kac representation of that system to analyze their fluctuations. Our precise analysis of the variance of a standard multilevel splitting algorithm reveals an opportunity for improvement. This leads to a novel method that relies on adaptive levels and produces, in the limit of an idealized version of the algorithm, estimates with optimal variance. The motivation for this theoretical work comes from problems occurring in watermarking and fingerprinting of digital contents, which represents a new field of applications of rare event simulation techniques. Some numerical results show performance close to the idealized version of our technique for these practical applications.

Keywords Rare event · Sequential importance sampling · Feynman-Kac formula · Metropolis-Hastings · Fingerprinting · Watermarking

This work was partially supported by the French Agence Nationale de la Recherche (ANR), project Nebbiano, number ANR-06-SETI-009.

F. Cérou (✉) · T. Furon
INRIA Rennes - Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France
e-mail: Frederic.Cerou@inria.fr

P. Del Moral
INRIA Bordeaux Sud-Ouest & Institut de Mathématiques de Bordeaux, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence Cedex, France

A. Guyader
Equipe de Statistique, Université de Haute Bretagne, Place du Recteur H. Le Moal, CS 24307, 35043 Rennes Cedex, France

1 Introduction

Monte Carlo approach is a common tool to estimate the expectation of any function of a random object when analytical or numerical methods are not available. However, if the function is non-zero on a set which has a very small probability, then the naive Monte Carlo method will return the estimate zero, unless the sample size is so large that it becomes intractable. Typically we want to estimate precisely and in a reasonable time the small probability, say 10^{-9} or below, of an extreme event. A naive Monte Carlo method is impractical as it would require an excessively large sample.

To circumvent this difficulty, *importance sampling* (see e.g. Bucklew 2004) changes the law of the simulated random objects, and reweighs the sample consequently. The difficulty is then to choose the appropriate change of probability in order to achieve a good estimate. This is not always obvious, especially when there is no large deviation result to consider.

Importance splitting or *multilevel splitting* is another approach that is well adapted when the random object is a Markovian process. The basic idea is to reinforce trajectories that approach the targeted set by splitting (or branching) them and discarding the others. This very powerful approach in fact dates back to Kahn and Harris (1951) and Rosenbluth and Rosenbluth (1955). We refer the reader to the paper by Glasserman et al. (1999) which contains a precise review on these methods as well as a detailed list of references. Recently, the connection between importance splitting for Markovian processes and particle methods for Feynman-Kac models has lead to some improvements and to a rigorous framework for mathematical analysis (see Del Moral 2004).

Unlike most of the previous works concerning rare event estimation and simulation, the present work deals with rare

events for a *fixed* probability distribution. We are simply concerned with events of the type $\{X \in A\}$ for some random vector X, with $p = \mathbb{P}(X \in A) = \mathbb{P}(\Phi(X) > L) \ll 1$, where Φ is a mapping from $\mathbb{R}^d$ to $\mathbb{R}$, and where there is no dynamical model for X, i.e. X is not a process indexed by the time. In order to use the framework developed for Markov processes (see Cérou et al. 2006; Cérou and Guyader 2007; Del Moral and Lezaud 2006), we construct a family of Markov transition kernels M_k whose invariant measures are the successive laws of X restricted on smaller and smaller sets, the smallest being A. As usual when using a splitting technique in rare event simulation, we decompose the rare event in not so rare nested events, whose product of probabilities equals the probability of the rare event.

To our knowledge, the first instance in which static rare event simulation using splitting was proposed is Au and Beck (2001) (see also Au and Beck 2003). But Au and Beck call it "Subset simulation" and do not make any connection with splitting, which is why people in the rare event community do not mention this work afterwards.

In the standard rare event literature, Del Moral et al. (2006) and Johansen et al. (2006) were first to use fixed-levels algorithms for static rare events. These articles were written in a different framework, and thus do not deal with the practical details of our precise setting. In the present article, we detail both a fixed and an adaptive multilevels algorithm, the adaptive one consisting in optimally placing the levels on the fly.

Botev and Kroese (2008) work on a similar algorithm, including the use of quantiles of the random variable $\Phi(X)$ on the swarm of particles in order to estimate the next level. The main difference is their two stage procedure (like in Garvels 2000): they first run the algorithm just to compute the levels, and then they restart from the beginning with these proposed levels. Actually we prove that by computing the levels on the fly (within the same run as the one to compute the rare event probability), we only pay a small bias on the estimate. Note also that Botev and Kroese (2008) does not address the general construction of the transition kernels M_k, since the authors only tackle examples where they can derive a Gibbs sampler at each step. This is mainly possible because their function Φ is linear, which is a severe restriction.

Another related approach is the recent work on combinatorial counting of Rubinstein (2008). This article presents some optimizations for counting problems in which X has a uniform distribution over a discrete but very large state space. The author uses what he calls a cloning procedure, where the number of offspring is fixed (i.e. the same for all the particles in the sample) but adaptive to keep the number roughly constant, while removing redundant particles after the MCMC step. This is a main difference since we use a resampling with replacement procedure. But clearly results in Rubinstein (2008) show that the adaptive procedure is well suited for SAT problems, or other hard finite set optimization problems. We would also like to mention that these last two papers (Botev and Kroese 2008; Rubinstein 2008) have demonstrated the performance of their algorithms via an extensive simulation study, to which we now lay out the mathematical foundations.

Finally, Botev and Kroese (2011) very recently revisited the different variants of splitting procedures for static rare event simulation. They also provide a nice discussion about a test for the stationarity of the MCMC step. They mainly claim that from a practical point of view, one should favor the variants without bias in the desired estimates. Although this is arguably the best choice for some applications, we think that a more precise theoretical study of the other (biased) versions can be of interest even for practitioners in other application areas.

The paper is organized as follows. Section 2 describes and analyses the fixed-levels algorithm. Section 3 provides the adaptive levels version, and theoretically analyzes an *idealized* version, which proves to be optimal in terms of asymptotic variance of the estimator. Section 4 deals with the tuning of the algorithm and especially the choice and the iteration of the transition kernel which is at the core of the method. Section 5 shows the relevance of our algorithm for watermarking and fingerprinting, which constitute a new application area of rare event simulation techniques. These numerical results also show that the performance of the idealized version is almost reached by the actual algorithm. Finally, all the proofs are gathered in Appendix.

2 The fixed-levels method

2.1 Assumptions and ingredients

We assume that X is a random vector on $\mathbb{R}^d$ for some $d > 0$, and denote by μ its probability distribution on the underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We denote by A the rare set of interest, and we assume that $A = \{x \in \mathbb{R}^d \text{s.t.} \Phi(x) \geq L\}$ for some function $\Phi : \mathbb{R}^d \mapsto \mathbb{R}$ and some real number L. We also assume that we know how to draw i.i.d. samples from μ.

Our algorithm makes use of the following ingredients. An increasing sequence $\{L_0, \ldots, L_n\}$ in $\overline{\mathbb{R}}$, with $L_0 = -\infty$ and $L_n = L$ defines a sequence of corresponding sets $A_k = \{x \in \mathbb{R}^d,\ \Phi(x) \geq L_k\}$. These sets are thus nested: $\mathbb{R}^d = A_0 \supset A_1 \supset \cdots \supset A_n = A$. We now need to choose sequence $\{L_0, \ldots, L_n\}$ in such a way that $p_k = \mathbb{P}(X \in A_{k+1} | X \in A_k)$ is not too small. For indices $k > n$, we assume that $L_k = L_n$. We also need to choose a Markov transition kernel K on $\mathbb{R}^d$ which is μ-symmetric, that is

$$\forall (x, y) \in \mathbb{R}^d \times \mathbb{R}^d,$$
$$\mu(dx)K(x, dy) = \mu(dy)K(y, dx).$$

As a consequence, K has μ as an invariant measure.

As we will see in the sequel, the choice of the L_k's can be made adaptive and is thus not an issue. However, the choice of the kernel K is crucial. Even if any μ-symmetric kernel would eventually do the job, we need to carefully choose it to make the algorithm efficient as discussed in Sect. 4.

Consider now a Markov chain $(X_k)_{k\geq 0}$ defined by: $\mathcal{L}(X_0) = \mu$ and the inhomogeneous transitions kernels $\mathbb{P}(X_k \in dy | X_{k-1} = x) = M_k(x, dy)$, with

$$M_k(x, dy) = \mathbb{1}_{A_k^c}(x)\delta_x(dy) + \mathbb{1}_{A_k}(x)(K(x, dy)\mathbb{1}_{A_k}(y) + K(x, A_k^c)\delta_x(dy)).$$

Moving a particle according to M_k is then twofold: firstly a new transition according to K is proposed, and secondly we accept this transition only if it stays in A_k, keeping the old position otherwise. For $k \in \{0, \ldots, n\}$, denote $\mu_k(dx) = \frac{1}{\mu(A_k)}\mathbb{1}_{A_k}(x)\mu(dx)$ the normalized restriction of μ on A_k.

We should also note at this point that instead of a μ-symmetric kernel K to construct the M_k, one can use at level k, any kernel, if available, for which μ_k is invariant. In some applications this can be done directly trough a Gibbs sampler (see Rubinstein 2008). We have chosen to adopt here a Metropolis-Hastings approach because it is somehow more general, and we will not particularly discuss this case. But from a practical point of view, if such a family of kernels M_k is readily available, then it is much advisable to use it.

The following stationarity property holds for μ and μ_k.

Proposition 1 *The measures μ and μ_k are both invariant by the transition kernel M_k.*

The proof is a straightforward computation and thus will be omitted.

From a general point of view, a Feynman-Kac representation for μ_k is a formula of the form

$$\mu_k(\varphi) = \frac{\mathbb{E}[\varphi(X_k)\prod_{m=0}^{k-1} G_m(X_m)]}{\mathbb{E}[\prod_{m=0}^{k-1} G_m(X_m)]},$$

where the potentials G_m are positive functions, and $(X_k)_{k\geq 0}$ is a non homogeneous Markov chain with transitions M_k. If we know how to draw realizations of the Markov chain, then we can compute $\mu_k(\varphi)$ with a Monte Carlo approach. But naive Monte Carlo is not efficient, because most of the realizations of the chain have small values for the product of the potentials. Anyway, in this form a much nicer Monte Carlo algorithm can be used. It mainly consists in keeping a cloud of particles (ξ_k^j), with time $0 \leq k \leq n$ and particle index $1 \leq j \leq N$. Then for each time step k, discard those with small potential G_k, and branch the others, with a rate proportional to $G_k(\xi_k^j)$. Then apply the Markov transition M_k to all the surviving particles, and iterate on the time step.

This approach has given birth to a huge amount of literature, and is often referred to as *Sequential Importance Sampling (SIS)* or *Sequential Monte Carlo (SMC)*. See the monograph by Del Moral (2004) for a theoretical overview and Doucet et al. (2001) for examples of applications. In our context, the Feynman-Kac representation for μ_k has the following form.

Proposition 2 *For every test function φ, for $k \in \{0, \ldots, n\}$, the Feynman-Kac representation is as follows*

$$\mu_k(\varphi) = \frac{\mathbb{E}[\varphi(X_k)\prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)]}{\mathbb{E}[\prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)]},$$

where $(X_k)_{k\geq 0}$ is a Markov chain given by the following conditions: $X_0 \sim \mu$ and the inhomogeneous transition kernels $(M_k)_{k\geq 1}$.

2.2 The fixed-levels algorithm

Proposition 2 shows that the framework of Feynman-Kac formulae does apply, and thus this grants access to the approximation of the associated measures using an interacting particle method as studied by Del Moral (2004). Basically, at each iteration k, it consists of selecting the particles according to the potentials, here $\mathbb{1}_{A_{k+1}}$, and then in propagating the particles according to the transitions given by M_{k+1}.

The approximation of the rare event probability stems from the following obvious property

$$p = \mathbb{P}(X \in A_n) = \prod_{k=0}^{n-1} \mathbb{P}(X \in A_{k+1} | X \in A_k) = \prod_{k=0}^{n-1} \mu_k(A_{k+1})$$

and finally

$$p = \prod_{k=0}^{n-1} \frac{\mathbb{E}[\mathbb{1}_{A_{k+1}}(X_k)\prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)]}{\mathbb{E}[\prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)]},$$

where the last equality comes from Proposition 2. We approximate at each stage $p_k = \mathbb{P}(X \in A_{k+1} | X \in A_k)$ by the proportion of the particles already in the next set, and the total probability is estimated as the product of those. This gives Algorithm 1.

Algorithm 1

Parameters

N the number of particles, the sequence $\{L_0, \ldots, L_n\}$ of levels.

Initialization

Draw an i.i.d. N-sample $(\xi_0^j)_{1\le j\le N}$, of the law μ.

Iterations

for $k=0$ to $n-1$ /* level number */
  Let $I_k=\{j:\ \xi_k^j\in A_{k+1}\}$.
  Let $\hat{p}_k=\frac{|I_k|}{N}$.
  for $j\in I_k$, let $\tilde{\xi}_{k+1}^j=\xi_k^j$
  for $j\notin I_k$, let $\tilde{\xi}_{k+1}^j$ be a copy of ξ_k^ℓ where ℓ is chosen randomly in I_k with uniform probabilities.
  for $j=1$ to N /* particle index */
    Draw a new particle $\hat{\xi}_{k+1}^j\sim K(\tilde{\xi}_{k+1}^j,.)$.
    If $\hat{\xi}_{k+1}^j\in A_{k+1}$ then let $\xi_{k+1}^j=\hat{\xi}_{k+1}^j$, else $\xi_{k+1}^j=\tilde{\xi}_{k+1}^j$.
  endfor
endfor

Output

Estimate the probability of the rare event by $\hat{p}=\prod_{k=0}^{n-1}\hat{p}_k$. The last set of particles is a (non independent) sample that provides an approximation of the law μ_n of the rare event.

2.3 Fluctuations analysis

Del Moral has extensively studied in a very general context the asymptotic behavior of the interacting particle model as the number N of particles goes to infinity (Del Moral 2004). For example, it is well known that the estimate $\hat{p}$ is unbiased. The next proposition presents a precise fluctuation result in our context of rare event analysis. It does not correspond exactly to Algorithm 1. The difference is that the proposition assumes that the resampling is done using a multinomial procedure, which gives a higher variance than that of Algorithm 1. This does not make much difference for the following discussion, as the best possible variance is the same. We have made this choice here because the terms of the variance for the multinomial procedure are a bit simpler to analyze and to relate to simple quantities characterizing the underlying Markov chains.

Proposition 3 *Let $\hat{p}$ denote the estimate given by the fixed-levels algorithm, then*

$$\sqrt{N}\,\frac{\hat{p}-p}{p}\xrightarrow[N\to+\infty]{\mathcal{D}}\mathcal{N}(0,\sigma^2),$$

with

$$\sigma^2=\sum_{k=0}^{n-1}\frac{1-p_k}{p_k}+\sum_{k=1}^{n-1}\frac{1}{p_k}\mathbb{E}\left[\left(\frac{\mathbb{P}(X_{n-1}\in A_n|X_k)}{\mathbb{P}(X_{n-1}\in A_n|X_{k-1}\in A_k)}-1\right)^2\middle|X_{k-1}\in A_k\right].$$

The samples are not independent due to the splitting of successful particles. In fact the variance is lower bounded

$$\sigma^2\ge\sum_{k=0}^{n-1}\frac{1-p_k}{p_k},$$

with equality if and only if for all $k=1,\dots,n-1$ and knowing that $X_{k-1}\in A_k$, one has

$$\mathbb{P}(X_{n-1}\in A_n|X_k)\perp X_k.$$

This means that equality holds if, between step k and step $n-1$, the algorithm forgets the initial position X_k. In order to reach this goal, a possible route is to begin step k by applying an infinite number of times (and not only one time as is the case in Algorithm 1) the transition kernel M_k with stationary distribution $\mu_k=\mathcal{L}(X|X\in A_k)$. We will discuss this point in detail in Sect. 4.

This will motivate us in the sequel to study an *idealized* version of the algorithm with at each step the possibility (never met in practice) to draw directly an i.i.d. sample of μ_k. As we will see from numerical results, the theoretical performance derived for this idealized version can almost be achieved by the actual algorithm at a reasonable cost.

Anyway, from now on, we *assume* that at each step k it is possible to draw an i.i.d. sample of the law of X conditionally on the event $\{X\in A_k\}=\{\Phi(X)>L_k\}$. Then the relative variance of the estimator reduces to:

$$\sigma^2=\sum_{k=0}^{n-1}\frac{1-p_k}{p_k}.$$

Thus, for a fixed value of p and a fixed number n of levels, this asymptotic variance would be minimal if $p_k\equiv p_0$ for all k. This is indeed a simple constrained optimization problem:

$$\arg\min_{p_0,\dots,p_{n-1}}\sum_{k=0}^{n-1}\frac{1-p_k}{p_k}\quad\text{s.t.}\quad\prod_{k=0}^{n-1}p_k=p.$$

In this case, the minimal asymptotic variance is simply $n\frac{1-p_0}{p_0}$, with $p_0=p^{\frac{1}{n}}$. This optimal situation corresponds to the case where the levels are evenly spaced in terms of probability of success: as far as multilevel splitting for Markov processes is concerned, this point was also mentioned in Glasserman et al. (1999), Lagnoux (2006) and Cérou et al. (2006). The following section addresses this crucial issue for the adaptive version of the algorithm. Before this, this section ends with two remarks.

Remarks

1. If one's particular interest is the variance of $\hat{p}$ rather than a convergence in distribution like the CLT-type result of Proposition 3, then we can turn to the recent

non asymptotic results obtained in Cérou et al. (2008, Corollary 5.2). Under some regularity conditions (mainly about the mixing property of the kernel K), there exist positive constants α_k, for $0 \le k \le n-1$, such that for all $N \ge N_0 = \sum_{k=0}^{n-1} \frac{\alpha_k}{p_k}$,

$$\mathbb{E}\left(\left[\frac{\hat{p}-p}{p}\right]^2\right) \le 4\frac{N_0}{N}.$$

If we assume an i.i.d. sample, then all the α_k's are all equal to 1, and $N_0 = \sum_{k=0}^{n-1} \frac{1}{p_k}$.

2. Finally, there is a maybe small, but non-zero, probability that the particle system dies at some stage. This may typically happen when two consecutive levels are too far apart, or when the number of particles is too small. A first solution to this problem is given in Le Gland and Oudjane (2006). The idea is to go on sampling new particles until a given number of them have reached the given level. The price to pay is a possibly very long computation time. A second solution is proposed in the next section.

3 The adaptive method

3.1 The algorithm

As we may not have a great insight about the law μ and/or the mapping Φ, typically when Φ is a 'black box', the choice of the levels $L_1, \dots, L_{n-1}$ might prove to be quite problematic. We propose from now on to adaptively choose the level sets, ensuring not only that the particle system never dies but also that the asymptotic variance of the estimate $\hat{p}$ is minimized.

The method is indeed simple to implement. We choose a prescribed success rate p_0 between two consecutive levels. In practice, $0.75 \le p_0 \le 0.8$ works well. At step k, the algorithm sorts the particles ξ_k^j according to their scores $\Phi(\xi_k^j)$. Then it sets the next level to the $(1-p_0)$ empirical quantile $\hat{L}_{k+1}$, which means that a proportion p_0 of the particles scores are above it. Starting from this sample of $p_0 N$ particles which are independently and identically distributed according to the law $\mathcal{L}(X|\Phi(X) > \hat{L}_{k+1})$, an i.i.d. sample of size N is drawn with the same distribution, and the rest of the algorithm is unchanged.

The algorithm then stops when some $\hat{L}_{\hat{n}_0+1} \ge L$, and the probability is estimated by $\hat{p} = \hat{r}_0\, p_0^{\hat{n}_0}$, where $\hat{r}_0$ denotes the number of particles in the last iteration being above level L. The number $\hat{n}_0$ of steps is random, but if N is large enough, then Lemma 1 in Appendix proves that, outside an event of exponentially small probability, $\hat{n}_0$ is actually fixed by the ratio of the logarithms

$$n_0 = \left\lfloor \frac{\log \mathbb{P}(X \in A)}{\log p_0} \right\rfloor = \left\lfloor \frac{\log p}{\log p_0} \right\rfloor. \tag{1}$$

As mentioned above, this variant enforces evenly spaced levels in terms of probability of success, and therefore a minimal asymptotic variance for the estimate $\hat{p}$ of p. The pseudo-code for the adaptive algorithm is given in Algorithm 2.

Algorithm 2

Parameters

N the number of particles, the number $N_0 < N$ of succeeding particles, and let $p_0 = N_0/N$.

Initialization

Draw an i.i.d. N-sample $(\xi_0^j)_{1\le j\le N}$ of the law μ.
Compute $\hat{L}_1$, the $(1-p_0)$ quantile of $\Phi(\xi_0^j)$,
$j = 1, \dots, N$.
$k = 1$;

Iterations

while $\hat{L}_k < L$ do
  Starting from an i.i.d. $p_0 N$-sample with law $\mathcal{L}(X|\Phi(X) > \hat{L}_k)$, draw an i.i.d. N-sample $(\xi_k^j)_{1\le j\le N}$ with the same law.
  Compute $\hat{L}_{k+1}$, the $(1-p_0)$ quantile of $\Phi(\xi_k^j)$, $j = 1, \dots, N$.
  $k = k+1$;
endwhile
Let N_L the number of particles ξ_{k-1}^j, $j = 1, \dots, N$, such that $\Phi(\xi_{k-1}^j) \ge L$.

Output

Estimate the probability of the rare event by $\hat{p} = \frac{N_L}{N} p_0^{k-1}$. The last set of particles is a (non independent) sample that provides an approximation of the law μ_n of the rare event.

Remarks

1. In this algorithm, the step drawing an N-sample starting from a $p_0 N$-sample is of course the trickiest one. The analytical study of this idealized version in the next subsection assumes it can be done perfectly, although this will never be met in practice. In Sect. 4, we propose a way to implement it in practice, at least approximately, and Sect. 5 shows its practical efficiency on two examples.
2. When X takes its values in a discrete space, it is likely that several values of $\Phi(\xi_k^j)$ are the same as the value of the $1-p_0$ quantile. In this case, one needs to be careful to count the exact number of particles N_k in $I_k = \{j : \Phi(\xi_k^j) \ge L_{k+1}\}$. And replace the last estimate by $\hat{p} = \prod N_k/N$.

3. The costs of adaptive levels is a higher complexity by a factor $\log N$ (due to the quick sort), and a slight loss of accuracy due to a bias. Yet, Proposition 4 proves that this bias becomes negligible compared to the standard deviation as N increases and provides an explicit formula, which allows to correct this bias and to derive confidence intervals. Experimental results of Sect. 5.3 illustrate this.
4. *Estimation of quantiles:* Some applications require the estimates of quantiles of the random variable $\Phi(X)$. This can be done at no additional cost within the previous algorithm. For $\alpha \in (0,1)$, define the α-quantile by $q_\alpha = \sup\{x : P(\Phi(X) \le x) \le \alpha\}$. When the algorithm is in step k, with a set of particles $\{\xi_k^j, j = 1, \dots, N\}$, such that $p_0^{k+1} \le 1-\alpha < p_0^k$, then let $r = 1-(1-\alpha)p^{-k}$. An estimate of the quantile q_α is then given by the r quantile of the sample $\{\Phi(\xi_k^j), j = 1, \dots, N\}$.

3.2 Bias and variance

The assumption of a continuous cumulative distribution function (cdf) F of $\Phi(X)$ is now required to derive the properties of the adaptive algorithm. Let us write the rare event probability as

$$p = r_0\, p_0^{n_0}, \quad \text{with } n_0 = \left\lfloor \frac{\log p}{\log p_0} \right\rfloor \quad \text{and} \quad r_0 = p\, p_0^{-n_0},$$

so that $r_0 \in (p_0, 1]$. In the same way we write $\hat{p} = \hat{r}_0\, p_0^{\hat{n}_0}$, with $\hat{n}_0$ the number of steps before the algorithm stops. A first theorem shows a CLT-type convergence.

Theorem 1 *If F is continuous, then we have*

$$\sqrt{N}\,(\hat{p} - p) \xrightarrow[N\to+\infty]{\mathcal{D}} \mathcal{N}(0, \sigma^2),$$

where

$$\sigma^2 = p^2 \left(n_0 \frac{1-p_0}{p_0} + \frac{1-r_0}{r_0} \right).$$

Unlike the fixed-levels version of the algorithm, the adaptive version is biased. Nevertheless, the next result shows that the bias is of order $1/N$, and is thus negligible compared to the standard deviation given in Theorem 1.

Proposition 4 *If F is continuous, then we have*

$$N \frac{\mathbb{E}[\hat{p}] - p}{p} \xrightarrow[N\to+\infty]{} n_0 \frac{1-p_0}{p_0}.$$

Thus the bias is positive and of order $\frac{1}{N}$ when N goes to infinity:

$$\frac{\mathbb{E}[\hat{p}] - p}{p} \sim \frac{1}{N} \frac{n_0(1-p_0)}{p_0}.$$

Putting all things together, we can write the following expansion:

$$\hat{p} = p\left(1 + \frac{1}{\sqrt{N}} \sqrt{n_0 \frac{1-p_0}{p_0} + \frac{1-r_0}{r_0}}\, Z + \frac{1}{N} n_0 \frac{1-p_0}{p_0} + o_{\mathbb{P}}\left(\frac{1}{N}\right)\right),$$

where Z is a standard Gaussian variable.

Remarks

1. The above formula shows that, although Algorithm 2 introduces a bias, it performs better than Algorithm 1 in terms of mean square error (bias included).
2. In this regard Remark 3.2, p. 489 of Botev and Kroese (2008) might be misleading: if the levels are chosen from start, or using a preliminary run, then the resulting probability estimate is unbiased, even if the level crossing probabilities are indeed dependent (see Cérou et al. 2006).
3. It is worth mentioning that the bias is always positive, giving a slightly overvalued estimate. As rare event analysis usually deals with catastrophic events, it is not a bad thing that the real value be a bit lower than the provided estimate. Moreover, if one wants to correct it, the explicit formula of Proposition 4 allows to do so.

4 Tuning the algorithm

4.1 Choice of the kernel K

There is no completely general method for finding the best transition kernel K because it depends on the application. But in the very classical case of a Gibbs measure given by a bounded potential, we can use the Metropolis algorithm, as first proposed by Metropolis et al. (1953), or the more general version later proposed by Hastings (1970).

4.2 Less dependent sample

As mentioned in Sect. 2.3, for the fixed-levels version of the algorithm we always have

$$\sigma^2 \ge \sum_{k=0}^{n-1} \frac{1-p_k}{p_k}.$$

The equality holds if and only if for all k, knowing that $X_{k-1} \in A_k$, one has

$$\mathbb{P}(X_{n-1} \in A_n | X_k) \perp X_k.$$

To reach this goal, a simple idea is to iterate the transition kernel M_k at each step as it provides more independence

among particles. This is well documented in the MCMC literature, e.g. Tierney (1994) in the case of Metropolis-Hastings kernels.

This means that the more the kernel is iterated, the closer (in distribution) we get to an independent sample. Thus, at each step of Algorithm 2 (adaptive levels), we can think of iterating the kernel a fixed number of times (for example 20 times in the simulations of Sect. 5.3).

4.3 Mixing property of the kernel K

We have written the algorithms using a unique kernel K for all the iterations. Usually it is quite easy to construct not only one, but a family of kernels that are all μ-symmetric, but with different mixing properties. This is useful when applying K to the current particles, most of the transitions are refused (their scores are below the current threshold). In this case, we propose a change to another kernel which is less mixing, i.e. in some way with "smaller steps", and thus with a lower probability of going below the current level L_k. On the other hand, when almost all the transitions are accepted, it means that the kernel is poorly mixing, and that we could decrease the variance by choosing a kernel K that is more mixing, i.e. with "larger step". For example, this is tuned by the parameter α in Sect. 5.3.

5 Applications

Our motivation comes from problems occurring in the protection of digital contents. Here the term watermarking refers to a set of techniques for embedding/hiding information in a digital file (typically audio or video), such that the change is not perceptible, and very hard to remove. See the web site of the Copy Protection Working Group Copy Protection Technical Working Group for details.

In order to be used in an application, a watermarking technique must be reliable. Here are two application scenarios where a wrong estimation of the probability of error could lead to a disaster.

5.1 Copy protection

Assume commercial contents are encrypted and watermarked and that future consumer electronics storage devices have a watermark detector. These devices refuse to record a watermarked content since it is copyrighted material. The probability of false alarm is the probability that the detector considers an original piece of content (which has not been watermarked) as protected. The movie that a user shot during his holidays could be rejected by his storage device. This absolutely non user-friendly behavior really scares consumer electronics manufacturers. In the past, the Copy Protection Working Group of the DVD forum evaluated that at most one false alarm should happen in 400 hours of video (Copy Protection Technical Working Group). As the detection rate was one decision per ten seconds, this implies a probability of false alarm in the order of 10^{-5}. An accurate experimental assessment of such a low probability of false alarm would demand to feed a real-time watermarking detector with non-watermarked content during 40,000 hours, i.e. more than 4 years! Proposals in response of the CPTWG's call were, at that time, never able to guarantee this level of reliability.

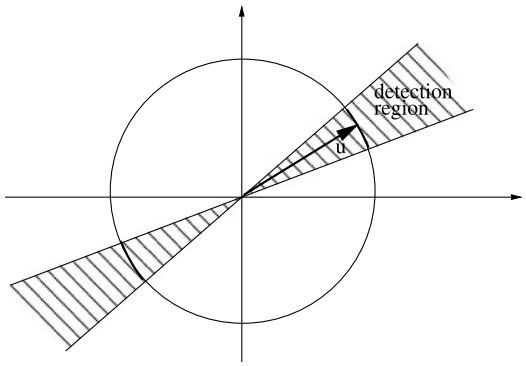

Fig. 1 Detection region for zero-bit watermarking

5.2 Fingerprinting

In this application, users' identifiers are embedded in a purchased content. When this content is found in an illegal place (e.g. a P2P network), the copyright holders decode the hidden message, find an identifier, and thus they can trace the traitor, i.e. the customer who has illegally broadcast his copy. However, the task is not that simple because dishonest users might collude. For security reason, anti-collusion codes have to be employed. Yet, these solutions (also called weak traceability codes, see Barg et al. 2003) have a non-zero probability of error (defined as the probability of accusing an innocent). This probability should be, of course, extremely low, but it is also a very sensitive parameter: anti-collusion codes get longer (in terms of the number of bits to be hidden in content) as the probability of error decreases. Fingerprint designers have to strike a trade-off, which is hard to conceive when only rough estimation of the probability of error is known. The major issue for fingerprinting algorithms is the fact that embedding large sequences implies also assessing reliability on a huge amount of data, which may be practically unachievable without using rare event analysis.

5.3 Zero-bit watermarking

In this example of zero-bit watermarking, X is a Gaussian vector in $\mathbb{R}^d$, with zero mean and identity covariance matrix,

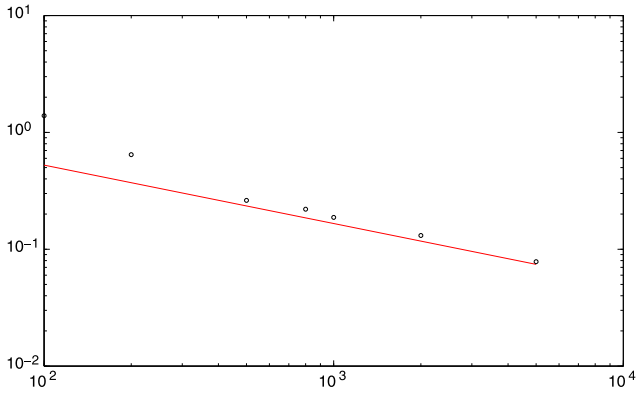

Fig. 2 Theoretical and empirical relative standard deviations with 100 simulations

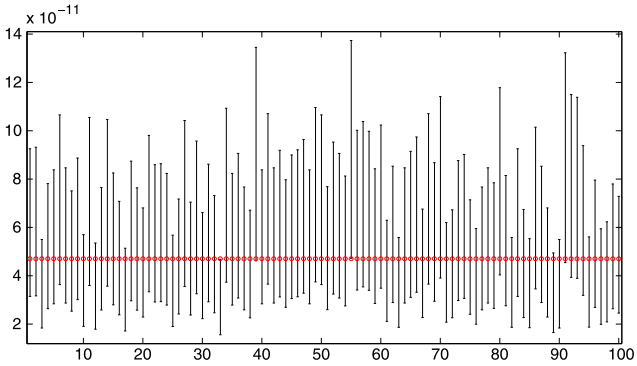

Fig. 3 95% confidence intervals for $p = 4.704 \cdot 10^{-11}$ with 100 simulations and $N = 500$ particles

$\Phi(X) = \frac{|\langle X,u\rangle|}{\|X\|}$ where u is a fixed normalized vector (see Merhav and Sabbag 2008). Then the region $A = \{x \in \mathbb{R}^d$ s.t. $\Phi(x) \geq L\}$ is a double hypercone as shown in Fig. 1. For a Gaussian distribution, the obvious choice for the kernel is the following: if we start from any point x, then the new position is given by

$$x' = \frac{x + \alpha W}{\sqrt{1+\alpha^2}},$$

where W is a $\mathcal{N}(0, I_d)$ $\mathbb{R}^d$ valued random vector and α a positive number.

This simple setup allows us to compare our estimates of the rare event probability with the result of a numerical integration. For example, in our simulations $d = 20$ and $L = 0.95$, so that $p = \mathbb{P}(\Phi(X) \geq L) \approx 4.70 \cdot 10^{-11}$ (tail probability of a Fisher distribution). For such a low probability, it is of course out of question to run a classical Monte Carlo algorithm. Our algorithm with adaptive levels was run with 20 iterations of the kernel M_k at each step. The choice of the mixing parameter $\alpha = 0.3$ has experimentally proved to be a good trade-off, see discussion in Sect. 4.3 above for details. The proportion of particles surviving from one step to another has been fixed to $p_0 = 0.75$, so that

$$n_0 = \left\lfloor \frac{\log p}{\log p_0} \right\rfloor = 82 \quad \text{and} \quad r_0 = p p_0^{-n_0} \approx 0.83.$$

For several numbers of particles, ranging from $N = 100$ to $N = 5{,}000$, we have run the algorithm 100 times in order to estimate the variance. Figure 2 shows in log-log plots the convergence of the normalized standard deviation to minimum achievable, which is that of i.i.d. samples at each stage, that is (see Theorem 1):

$$\sqrt{\mathrm{Var}\left(\frac{\hat{p}-p}{p}\right)} \sim \frac{1}{\sqrt{N}} \sqrt{n_0 \frac{1-p_0}{p_0} + \frac{1-r_0}{r_0}}.$$

Clearly this indicates that even if we use the mixing kernel a finite number of times (here only 20 times), the empirical variance is on the same level as it would be for an independent sample (that is in the limit of infinite applications of the kernel), and therefore our theoretical results in this setting give a good picture of the accuracy of the actual algorithm.

Anyway, from a practical point of view, one would like to obtain an estimation of p by running only one time the algorithm. In this aim, Theorem 1 and Proposition 4 allow us to construct confidence intervals. Indeed, we have

$$\frac{\hat{p}-p}{p} \approx \mathcal{N}\left(\frac{n_0(1-p_0)/p_0}{N}, \frac{n_0(1-p_0)/p_0 + (1-r_0)/r_0}{N}\right),$$

so that an approximate 95% confidence interval for p is given by $I = [\hat{p}_-, \hat{p}^+]$, where

$$\hat{p}_-^+ = \hat{p}\left(1 - \frac{\hat{n}_0(1-p_0)/p_0}{N} \pm 2\sqrt{\frac{\hat{n}_0(1-p_0)/p_0 + (1-\hat{r}_0)/\hat{r}_0}{N}}\right).$$

This is illustrated in Fig. 3, where 100 such confidence intervals have been drawn for $N = 500$ particles. In this example, 2 of them do not contain the true value $p = 4.704 \cdot 10^{-11}$. Once again, we would like to emphasize that the explicit formula for the bias (cf. Proposition 4) allows to cancel it, and consequently that this existence of a bias in the adaptive method is not a problem at all.

5.4 Tardos probabilistic codes

We are interested here in embedding an identifier in each copy of a purchased content. Then a copy, which is the result of a collusion, is found on the web, and we want to decide whether or not it can be originated from a certain user. The rare event will be to consider an innocent as guilty.

The embedded message, called a fingerprint, consists of a sequence of bits $X = (X_1, \ldots, X_m)$, where each X_i is independent from the others, and drawn from a Bernoulli's

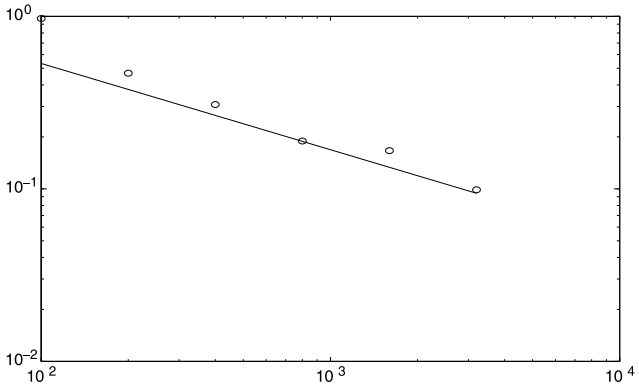

Fig. 4 Theoretical and empirical relative standard deviations with 100 simulations for an example of Tardos code

$\mathcal{B}(p_i)$. The p_i's are themselves i.i.d. random variables, drawn from a given distribution with density f on $[0, 1]$. Then we find a copy with fingerprint $y = (y_1, \ldots, y_m) \in \{0, 1\}^m$. We conclude that a user is guilty if the score

$$\Phi(X) = \sum_{i=1}^{m} y_i g_i(X_i)$$

is larger than some value L, for some given functions g_i's. This approach was proposed by Tardos (2003), where he derives optimal choices for f and the g_i's.

To apply our algorithm, we need to choose the kernel K. As the X_i's are independent, we randomly choose r indices $\{j_1, \ldots, j_r\} \in \{1, \ldots, m\}$, with r being a fixed parameter. Then for each j_ℓ, we draw a new X'_{j_ℓ} independently from the Bernoulli distribution $\mathcal{B}(p_{j_\ell})$.

For such codes, we first present the equivalent of Fig. 2 in Fig. 4. We consider the probability of accusing an innocent using a code of length $m = 200$. The algorithm was run with $p_0 = 1/2$. As we do not have any other estimates on the rare event probability, we just plugged the mean of the estimates given by the runs of our algorithm with the largest number of particles (3200) in the theoretical variance given by Theorem 1. This best estimate on the probability of the rare event is 2.6×10^{-9}. Again, with only 20 applications of the kernel at each iteration, we see that the performance of the algorithm is close to that of the idealized version. It is noticeable that this remains true even if in this case the assumption on the cdf F of $\Phi(X)$ of Sect. 3.2 is clearly not fulfilled. Figure 5 shows the distribution of the number of steps as a function of the number of particles. We can see that for 800 particles and more, the number of steps can be seen practically as deterministic. All these results illustrate the efficiency of our algorithm for discrete problems.

Using our adaptive algorithm, we made some additional numerical experiments on such codes. More precisely, we can easily estimate the probability of false detection (false positive) for some code length m, and collusion size c.

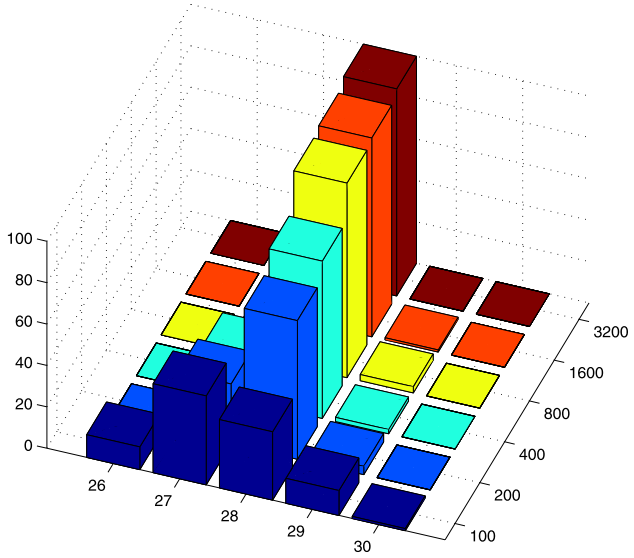

Fig. 5 Distribution of the number of steps

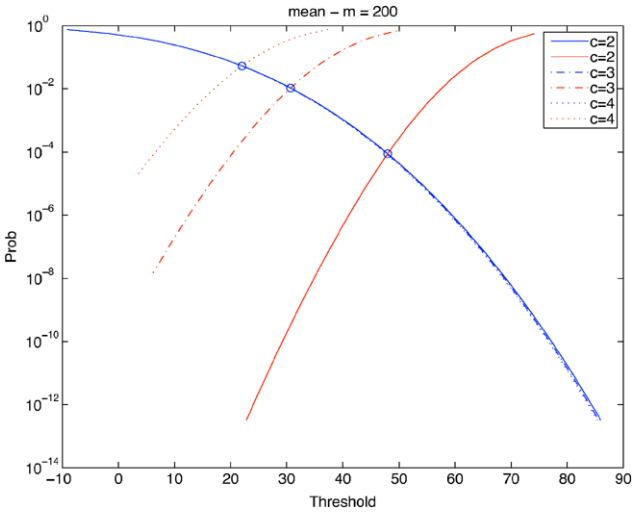

Fig. 6 (Color online) Mappings of the false positive probability (*blue*) and false negative probability (*red*) against the threshold. $m = 200$, $c \in \{2, 3, 4\}$. The score of a particle is the mean of the c colluders scores

The collusion strategy is to randomly pick up the symbols of pirated copy among the c colluders' sequences. We can also estimate the probability of not accusing someone guilty (false negative). The results for $m = 200$, and $c = 2, 3, 4$ are shown in Fig. 6. From these curves, one can then decide how to set the threshold L to minimize the total error.

Appendix: Proofs

Proof of Proposition 2 We use induction to show that

$$\mathbb{E}\left[\varphi(X_k) \prod_{m=0}^{k-1} \mathbb{1}_{A_{m+1}}(X_m)\right] = \mu(A_k)\, \mu_k(\varphi).$$

The case $k=0$ is obvious. Then assume the property be true for k. We write, using the Markov property and Proposition 1,

$$\begin{aligned}&\mathbb{E}\left[\varphi(X_{k+1})\prod_{m=0}^{k}\mathbb{1}_{A_{m+1}}(X_m)\right]\\&\quad=\mu_k(M_{k+1}(\varphi)\mathbb{1}_{A_{k+1}})\times\mu(A_k)\\&\quad=\mu_{k+1}(M_{k+1}(\varphi))\times\mu(A_{k+1})\\&\quad=\mu_{k+1}(\varphi)\times\mu(A_{k+1}).\end{aligned}$$

Then taking the case $\varphi=\mathbb{1}$ we have $\mathbb{E}[\prod_{m=0}^{k}\mathbb{1}_{A_{m+1}}(X_m)]=\mu(A_{k+1})$, which concludes the proof. □

Proof of Proposition 3 Adopting the notation of Proposition 9.4.1, p. 301 of Del Moral (2004), the application of the first formula of p. 304 leads to

$$\begin{aligned}\mathrm{Var}(\hat{p})&=\mathbb{E}[W_{n-1}^{\gamma}(1)^2]\\&=\gamma_{n-1}^2\sum_{k=0}^{n-1}\eta_k\left(\left(\frac{Q_{k,n-1}(1)}{\eta_k Q_{k,n-1}(1)}-1\right)^2\right).\end{aligned}$$

In our context this can be rewritten as follows

$$\begin{aligned}\frac{\mathrm{Var}(\hat{p})}{p^2}=\sum_{k=0}^{n-1}\mathbb{E}\Bigg[\Bigg(&\frac{\mathbb{P}(X_{n-1}\in A_n|X_k)}{\mathbb{P}(X_{n-1}\in A_n|X_{k-1}\in A_k)}\\&-1\Bigg)^2\Bigg|X_{k-1}\in A_k\Bigg]\end{aligned}\tag{2}$$

where by convention $X_{-1}=X_0$. This leads to

$$\frac{\mathrm{Var}(\hat{p})}{p^2}=\sum_{k=0}^{n-1}\left(\frac{\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_{k-1}\in A_k]}{\mathbb{P}(X_{n-1}\in A_n)^2}-1\right).$$

Now one can readily check that

$$\begin{aligned}&\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_{k-1}\in A_k]\\&\quad=\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2\mathbb{1}_{A_{k+1}}(X_k)|X_{k-1}\in A_k],\end{aligned}$$

and this is equivalent to

$$\begin{aligned}&\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_{k-1}\in A_k]\\&\quad=\frac{\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2\mathbb{1}_{A_{k+1}}(X_k)\mathbb{1}_{A_k}(X_{k-1})]}{\mathbb{P}(X_{k-1}\in A_k)}.\end{aligned}$$

Since $\{X_k\in A_{k+1}\}$ implies $\{X_{k-1}\in A_k\}$, this last expression can be simplified

$$\begin{aligned}&\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_{k-1}\in A_k]\\&\quad=\frac{\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2\mathbb{1}_{A_{k+1}}(X_k)]}{\mathbb{P}(X_{k-1}\in A_k)},\end{aligned}$$

and rewritten as

$$\begin{aligned}&\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_{k-1}\in A_k]\\&\quad=\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_k\in A_{k+1}]\frac{\mathbb{P}(X_k\in A_{k+1})}{\mathbb{P}(X_{k-1}\in A_k)}.\end{aligned}$$

Since $p_k=\mathbb{P}(X_k\in A_{k+1}|X_{k-1}\in A_k)$ and $\{X_k\in A_{k+1}\}\Rightarrow\{X_{k-1}\in A_k\}$, we deduce

$$\begin{aligned}&\mathbb{P}(X_{n-1}\in A_n|X_{k-1}\in A_k)\\&\quad=p_k\,\mathbb{P}(X_{n-1}\in A_n|X_k\in A_{k+1}),\end{aligned}$$

so that

$$\begin{aligned}&\frac{\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_{k-1}\in A_k]}{\mathbb{P}(X_{n-1}\in A_n|X_{k-1}\in A_k)^2}\\&=\frac{\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_k\in A_{k+1}]}{p_k\,\mathbb{P}(X_{n-1}\in A_n|X_k\in A_{k+1})^2}.\end{aligned}$$

Now it remains to notice that

$$\begin{aligned}&\frac{\mathbb{E}[\mathbb{P}(X_{n-1}\in A_n|X_k)^2|X_k\in A_{k+1}]}{\mathbb{P}(X_{n-1}\in A_n|X_k\in A_{k+1})^2}=\\&1+\mathbb{E}\left[\left(\frac{\mathbb{P}(X_{n-1}\in A_n|X_k)}{\mathbb{P}(X_{n-1}\in A_n|X_k\in A_{k+1})}-1\right)^2\middle|X_k\in A_{k+1}\right],\end{aligned}$$

so that coming back to (2) gives the desired result

$$\begin{aligned}\frac{\mathrm{Var}(\hat{p})}{p^2}=&\sum_{k=0}^{n-1}\frac{1-p_k}{p_k}\\&+\sum_{k=0}^{n-1}\tfrac{1}{p_k}\mathbb{E}\left[\left(\tfrac{\mathbb{P}(X_{n-1}\in A_n|X_k)}{\mathbb{P}(X_{n-1}\in A_n|X_{k-1}\in A_k)}-1\right)^2\middle|X_{k-1}\in A_k\right],\end{aligned}$$

where the first term (i.e., for $k=0$) of the second sum is equal to zero since by convention $X_{-1}=X_0$. □

Proof of Theorem 1 In order to simplify the writings, we will suppose that p_0N is an integer. Then, for all real numbers L and L' such that $L<L'$, let us denote

$$\begin{aligned}F(L,L')&=\mathbb{P}(\Phi(X)\le L'|\Phi(X)>L)\\&=\frac{F(L')-F(L)}{1-F(L)},\end{aligned}$$

with the convention that $F(L,L')=0$ if $F(L)=1$.

We first notice the following crucial point: given $\hat{L}_k$, the random vectors ξ_k^j, for $j\in\{1,\dots,N\}$, are i.i.d., and thus so are the random variables $\Phi(\xi_k^j)$. Since F is continuous, given $\hat{L}_k$, the random variable $F(\hat{L}_k,\hat{L}_{k+1})$ has the same distribution as the random variable $U_{((1-p_0)N)}$, where $(U_i)_{1\le i\le N}$ is a sample of i.i.d. random variables with uniform law on $[0,1]$, and for all $N\ge1$

$$U_{(1)}\le U_{(2)}\le\cdots\le U_{(N)}.$$

Let us denote G_N the empirical cdf of $(U_i)_{1\le i\le N}$ and $G(x)=x$ the cdf of the uniform law on $[0,1]$. Then from the basic identities $\|G_N - G\|_\infty = \|G_N^{-1} - G\|_\infty$ and $U_{((1-p_0)N)} = G_N^{-1}(1-p_0)$, we can deduce that

$$|U_{((1-p_0)N)} - (1-p_0)| \le \|G_N - G\|_\infty.$$

Using Dvoretzky-Kiefer-Wolfowitz (DKW) inequality (see for example van der Vaart 1998), we have then for all $\varepsilon > 0$

$$\mathbb{P}(\|G_N - G\|_\infty > \varepsilon) \le 2\exp(-2N\varepsilon^2),$$

hence

$$\begin{aligned}&\mathbb{P}(|F(\hat{L}_k,\hat{L}_{k+1}) - (1-p_0)| > \varepsilon)\\ &\quad = \mathbb{P}(|U_{((1-p_0)N)} - (1-p_0)| > \varepsilon)\\ &\quad \le 2\exp(-2N\varepsilon^2).\end{aligned}$$

Using Borel-Cantelli lemma, we conclude that

$$F(\hat{L}_k,\hat{L}_{k+1}) \xrightarrow[N\to\infty]{a.s.} 1-p_0. \tag{3}$$

From the theory of order statistics (see for example Arnold et al. 1992, Theorem 8.5.1, p. 223), we also deduce the convergence in distribution

$$\sqrt{N}(1 - F(\hat{L}_k,\hat{L}_{k+1}) - p_0) \xrightarrow[N\to\infty]{\mathcal{D}} \mathcal{N}(0, p_0(1-p_0)). \tag{4}$$

To prove the result of Theorem 1, we proceed by induction, assuming that:

$$\sqrt{N}\left(\prod_{m=1}^{k}(1 - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^k\right) \xrightarrow[N\to\infty]{\mathcal{D}} \mathcal{N}(0,\sigma_k^2).$$

For the next step, we use the decomposition

$$\begin{aligned}&\sqrt{N}\left(\prod_{m=1}^{k+1}(1 - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^{k+1}\right)\\ &\quad = \sqrt{N}\left(\prod_{m=1}^{k}(1 - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^k\right)\\ &\qquad \times (1 - F(\hat{L}_k,\hat{L}_{k+1}) - p_0)\\ &\qquad + p_0\sqrt{N}\left(\prod_{m=1}^{k}(1 - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^k\right)\\ &\qquad + p_0^k\sqrt{N}(1 - F(\hat{L}_k,\hat{L}_{k+1}) - p_0).\end{aligned} \tag{5}$$

The almost sure convergence of (3) and the induction hypothesis ensure that the first term converges in probability to 0 when N goes to infinity. To prove the convergence in distribution of the other terms of (5), let us introduce the characteristic function

$$\begin{aligned}\phi_N(t) = \mathbb{E}\Bigg[\exp\Bigg(it\Bigg(p_0\sqrt{N}\Bigg(\prod_{m=1}^{k}(1 - F(\hat{L}_{m-1},\hat{L}_m))\\ - p_0^k\Bigg) + p_0^k\sqrt{N}(1 - F(\hat{L}_k,\hat{L}_{k+1}) - p_0)\Bigg)\Bigg)\Bigg].\end{aligned}$$

Conditioning with respect to $\hat{L}_1,\dots,\hat{L}_k$ gives

$$\begin{aligned}\phi_N(t) = \mathbb{E}\Bigg[\exp\Bigg(it\Bigg(p_0\sqrt{N}\Bigg(\prod_{m=1}^{k}(1\\ - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^k\Bigg)\Bigg)\Bigg)\\ \times \mathbb{E}[\exp(it(p_0^k\sqrt{N}(1 - F(\hat{L}_k,\hat{L}_{k+1})\\ - p_0))|\hat{L}_1,\dots,\hat{L}_k]\Bigg].\end{aligned}$$

Thanks to the strong Markov property of the $\hat{L}_k$'s, this can be reduced to

$$\begin{aligned}\phi_N(t) = \mathbb{E}\Bigg[\exp\Bigg(it\Bigg(+p_0\sqrt{N}\Bigg(\prod_{m=1}^{k}(1\\ - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^k\Bigg)\Bigg)\Bigg)\\ \times \mathbb{E}[\exp(it(p_0^k\sqrt{N}(1 - F(\hat{L}_k,\hat{L}_{k+1})\\ - p_0))|\hat{L}_k]\Bigg],\end{aligned}$$

and we can remark that

$$\begin{aligned}&\mathbb{E}[\exp(it(p_0^k\sqrt{N}(1 - F(\hat{L}_k,\hat{L}_{k+1}) - p_0))|\hat{L}_k]\\ &\quad = \mathbb{E}[\exp(it(p_0^k\sqrt{N}(1 - U_{((1-p_0)N)} - p_0))],\end{aligned}$$

where $U_{((1-p_0)N)}$ is independent of $\hat{L}_1,\dots,\hat{L}_k$. This leads to

$$\begin{aligned}\phi_N(t) = \mathbb{E}\Bigg[\exp\Bigg(it\Bigg(+p_0\sqrt{N}\Bigg(\prod_{m=1}^{k}(1\\ - F(\hat{L}_{m-1},\hat{L}_m)) - p_0^k\Bigg)\Bigg)\Bigg)\Bigg]\\ \times \mathbb{E}[\exp(it(p_0^k\sqrt{N}(1 - U_{((1-p_0)N)} - p_0))].\end{aligned}$$

Thanks to the induction hypothesis and to (4), it comes

$$p_0\sqrt{N}\left(\prod_{m=1}^{k}(1-F(\hat{L}_{m-1},\hat{L}_m))-p_0^k\right)$$
$$+\,p_0^k\sqrt{N}(1-F(\hat{L}_k,\hat{L}_{k+1})-p_0)$$
$$\xrightarrow[N\to\infty]{\mathcal{D}}\mathcal{N}(0,p_0^2\sigma_k^2+p_0^{2k+1}(1-p_0)).$$

Putting all pieces together gives finally

$$\sqrt{N}\left(\prod_{m=1}^{k+1}(1-F(\hat{L}_{m-1},\hat{L}_m))-p_0^{k+1}\right)$$
$$\xrightarrow[N\to\infty]{\mathcal{D}}\mathcal{N}(0,\sigma_{k+1}^2) \tag{6}$$

with $\sigma_{k+1}^2=p_0^2\sigma_k^2+p_0^{2k+1}(1-p_0)$. From this recursion we deduce that for all $k\geq 1$

$$\sigma_k^2=kp_0^{2k-1}(1-p_0).$$

It remains to deal with the last step. For the sake of simplicity, we suppose that $\log p/\log p_0$ is not an integer. Let us first consider an alternative algorithm defined as follows: we run Algorithm 2 with the deterministic number of steps n_0 and denote $\hat{p}_d=\hat{r}_d\ p_0^{n_0}$ the corresponding estimator, where

$$\hat{r}_d=\frac{1}{N}\sum_{j=1}^{N}\mathbb{1}_{\{\Phi(\xi_{n_0}^j)\geq L\}}. \tag{7}$$

In this expression, knowing $\hat{L}_{n_0}$, the random variables $(\mathbb{1}_{\{\Phi(\xi_{n_0}^j)\geq L\}})_{1\leq j\leq N}$ are i.i.d. Bernoulli trials with parameter

$$r=\mathbb{P}(\mathbb{1}_{\{\Phi(\xi_{n_0}^j)\geq L\}}=1|\hat{L}_{n_0})$$
$$=1-F(\hat{L}_{n_0},L)=\frac{p}{1-F(\hat{L}_{n_0})}=\frac{r_0p_0^{n_0}}{1-F(\hat{L}_{n_0})}.$$

Then we can write

$$\sqrt{N}(r_0-r)=\sqrt{N}(1-F(\hat{L}_{n_0})-p_0^{n_0})\frac{r_0}{1-F(\hat{L}_{n_0})}.$$

Now we use the almost sure convergence

$$1-F(\hat{L}_{n_0})\xrightarrow[N\to\infty]{a.s.}p_0^{n_0},$$

and the convergence in distribution from (6)

$$\sqrt{N}(1-F(\hat{L}_{n_0})-p_0^{n_0})\xrightarrow[N\to\infty]{\mathcal{D}}\mathcal{N}(0,\sigma_{n_0}^2),$$

to conclude that

$$\sqrt{N}(r_0-r)\xrightarrow[N\to\infty]{\mathcal{D}}\mathcal{N}\left(0,n_0\frac{1-p_0}{p_0}r_0^2\right).$$

From this we deduce that

$$\sqrt{N}(r_0-\hat{r}_d)\xrightarrow[N\to\infty]{\mathcal{D}}\mathcal{N}\left(0,n_0\frac{1-p_0}{p_0}r_0^2+\frac{1-r_0}{r_0}\right).$$

Since $\hat{p}_d=\hat{r}_d\ p_0^{n_0}$, it comes

$$\sqrt{N}(\hat{p}_d-p)\xrightarrow[N\to\infty]{\mathcal{D}}\mathcal{N}\left(0,p^2\left(n_0\frac{1-p_0}{p_0}+\frac{1-r_0}{r_0}\right)\right).$$

Coming back to the "true" estimator $\hat{p}$, we have

$$\sqrt{N}(\hat{p}-p)=\sqrt{N}(\hat{p}-p)\mathbb{1}_{\hat{n}_0=n_0}+\sqrt{N}(\hat{p}-p)\mathbb{1}_{\hat{n}_0\neq n_0},$$

but one can readily see that $\hat{p}\mathbb{1}_{\hat{n}_0=n_0}=\hat{p}_d\mathbb{1}_{\hat{n}_0=n_0}$ almost surely, so that

$$\sqrt{N}(\hat{p}-p)=\sqrt{N}(\hat{p}_d-p)+\sqrt{N}(\hat{p}-\hat{p}_d)\mathbb{1}_{\hat{n}_0\neq n_0},$$

and the proof of Theorem 1 will be complete if we show that

$$\sqrt{N}(\hat{p}-\hat{p}_d)\mathbb{1}_{\hat{n}_0\neq n_0}\xrightarrow[N\to\infty]{\mathbb{P}}0.$$

In this aim, let us first remark that for all $\varepsilon>0$

$$\mathbb{P}(|\sqrt{N}(p-\hat{p})\mathbb{1}_{\hat{n}_0\neq n_0}|>\varepsilon)\leq\mathbb{P}(\hat{n}_0\neq n_0),$$

then the following lemma allows us to conclude. □

Lemma 1 *Denoting* $c=\min(p_0-p^{\frac{1}{n_0}},p^{\frac{1}{n_0+1}}-p_0)$, *we have*

$$\mathbb{P}(\hat{n}_0\neq n_0)\leq 2(n_0+1)e^{-2Nc^2}.$$

As a consequence

$$\hat{n}_0\xrightarrow[N\to\infty]{a.s.}n_0.$$

Proof Let us denote $B=\{\hat{n}_0=n_0\}$ the event for which the algorithm stops after the correct number of steps. The following equalities are straightforward:

$$B=\{\hat{L}_{n_0}\leq L<\hat{L}_{n_0+1}\}$$
$$=\{1-F(\hat{L}_{n_0+1})<1-F(L)\leq 1-F(\hat{L}_{n_0})\}$$
$$=\left\{\prod_{m=1}^{n_0+1}(1-F(\hat{L}_{m-1},\hat{L}_m))<p\right.$$
$$\left.\leq\prod_{m=1}^{n_0}(1-F(\hat{L}_{m-1},\hat{L}_m))\right\}.$$

For all $m=1,\ldots,n_0+1$, if we denote

$$B_m=\{p^{\frac{1}{n_0}}-p_0<1-p_0-F(\hat{L}_{m-1},\hat{L}_m)<p^{\frac{1}{n_0+1}}-p_0\},$$

we have

$$\mathbb{P}(B) \geq \mathbb{P}(B_1 \cap \cdots \cap B_{n_0+1}) \geq 1 - \sum_{m=1}^{n_0+1} (1 - \mathbb{P}(B_m)).$$

Denoting $c = \min(p_0 - p^{\frac{1}{n_0}}, p^{\frac{1}{n_0+1}} - p_0)$, the DKW inequality implies

$$1 - \mathbb{P}(B_m) \leq \mathbb{P}(|1 - p_0 - F(\hat{L}_{m-1}, \hat{L}_m)| > c) \leq 2e^{-2Nc^2},$$

so that the result of Lemma 1 is proved

$$\mathbb{P}(B) = \mathbb{P}(\hat{n}_0 = n_0) \geq 1 - 2(n_0 + 1)e^{-2Nc^2}.$$

□

Proof of Proposition 4 As for the analysis of the standard deviation, we first notice that

$$N(\hat{p} - p) = N(\hat{p}_d - p) + N(\hat{p} - \hat{p}_d)\mathbb{1}_{\hat{n}_0 \neq n_0}, \tag{8}$$

and applying Lemma 1 yields

$$N\mathbb{E}[|\hat{p} - \hat{p}_d|\mathbb{1}_{\hat{n}_0 \neq n_0}] \leq N\mathbb{P}(\hat{n}_0 \neq n_0) \underset{N\to\infty}{\longrightarrow} 0,$$

so that only the first term of the right-hand-side of (8) is worth considering for the convergence of $N\ \mathbb{E}[\hat{p} - p]$. Recall that the estimate is then $\hat{p}_d = \hat{r}_d\ p_0^{n_0}$, where $\hat{r}_d$ is defined as in (7), so that

$$\mathbb{E}[\hat{r}_d] = \mathbb{E}[\mathbb{E}[\hat{r}_d|\hat{L}_{n_0}]] = \mathbb{E}[r] = \mathbb{E}\left[\frac{p}{1 - F(\hat{L}_{n_0})}\right].$$

Then the normalized bias is

$$\begin{aligned}\frac{\mathbb{E}[\hat{p}] - p}{p} &= \mathbb{E}\left[\frac{p_0^{n_0}}{1 - F(\hat{L}_{n_0})}\right] - 1 \\ &= \mathbb{E}\left[\frac{F(\hat{L}_{n_0}) - F(L_{n_0})}{1 - F(\hat{L}_{n_0})}\right] \\ &= \mathbb{E}\left[\frac{W}{a - W}\right]\end{aligned}$$

with $W = F(\hat{L}_{n_0}) - F(L_{n_0}) = F(\hat{L}_{n_0}) - (1 - p_0^{n_0})$, and $a = 1 - F(L_{n_0}) = p_0^{n_0}$. If we remark that

$$1 - F(\hat{L}_{n_0}) = \prod_{k=0}^{n_0-1} (1 - F(\hat{L}_k, \hat{L}_{k+1}))$$

with the convention $\hat{L}_0 = -\infty$, then the result of (3) implies

$$\frac{W}{a} = \frac{F(\hat{L}_{n_0}) - (1 - p_0^{n_0})}{p_0^{n_0}} \underset{N\to+\infty}{\overset{a.s.}{\longrightarrow}} 0.$$

We may now rewrite

$$\frac{\mathbb{E}[\hat{p}] - p}{p} = \frac{1}{a}\mathbb{E}\left[W\frac{1}{1 - \frac{W}{a}}\right],$$

and the asymptotic expansion $(1-x)^{-1} = 1+x+o(x)$ leads to

$$\frac{\mathbb{E}[\hat{p}] - p}{p} = \frac{1}{a}\mathbb{E}[W] + \frac{1}{a^2}\mathbb{E}[W^2] + \frac{1}{a^2}o(\mathbb{E}[W^2]). \tag{9}$$

Then we will use the following lemma.

Lemma 2

$$\mathbb{E}[W] = \mathbb{E}[F(\hat{L}_{n_0}) - (1 - p_0^{n_0})] = 0$$

and

$$\begin{aligned}\mathbb{E}[W^2] &= \mathrm{Var}(F(\hat{L}_{n_0}) - F(L_{n_0})) \\ &= \frac{n_0}{N} p_0^{2n_0-1}(1 - p_0) + o\left(\frac{1}{N}\right).\end{aligned}$$

The proof of this lemma is left to the next subsection. Coming back to (9), we have obtained

$$\frac{\mathbb{E}[\hat{p}] - p}{p} = \frac{1}{N}\left(n_0\frac{1 - p_0}{p_0}\right) + o\left(\frac{1}{N}\right),$$

which ends the proof of Proposition 4. □

Proof of Lemma 2 First of all, some notation. Let $(U_i)_{1\leq i\leq N}$ be an i.i.d. family of random variables uniformly distributed on $(0, 1)$. We denote by $U_{(i)}$ the ith largest sample: $0 \leq U_{(1)} \leq \cdots \leq U_{(N)} \leq 1$. For simplicity, we will assume that $p_0 = \frac{N_0}{N}$ for some $1 \leq N_0 \leq N$. Then it is well known from the theory of order statistics (see for example e.g. Arnold et al. 1992 formula (2.2.20), p. 14) that

$$\mathbb{E}[U_{(N-N_0)}] = 1 - p_0. \tag{10}$$

Expectation of W We will prove that it is equal to zero by induction on n_0. For $n_0 = 1$, $F(\hat{L}_1)$ has the same law as $U_{(N-N_0)}$, thus the result is obvious by (10). Now assume that $\mathbb{E}[F(\hat{L}_{n_0-1})] = 1 - p_0^{n_0-1}$. From the decomposition

$$\prod_{k=1}^{n_0} (1 - F(\hat{L}_{k-1}, \hat{L}_k)) = 1 - F(\hat{L}_{n_0}),$$

we deduce

$$\begin{aligned}\mathbb{E}[1 - F(\hat{L}_{n_0})] &= \mathbb{E}\left[\prod_{k=1}^{n_0} (1 - F(\hat{L}_{k-1}, \hat{L}_k))\right] \\ &= \mathbb{E}\Big[\mathbb{E}[1 - F(\hat{L}_{n_0-1}, \hat{L}_{n_0})|\hat{L}_{n_0-1}]\end{aligned}$$

$$\times \prod_{k=1}^{n_0-1} (1 - F(\hat{L}_{k-1}, \hat{L}_k)) \Bigg].$$

Since

$$\mathbb{E}[1 - F(\hat{L}_{n_0-1}, \hat{L}_{n_0}) | \hat{L}_{n_0-1}] = \mathbb{E}[1 - U_{(N-N_0)}] = p_0,$$

the induction property for $n_0 - 1$ implies

$$\mathbb{E}[1 - F(\hat{L}_{n_0})] = p_0 \mathbb{E}\left[\prod_{k=1}^{n_0-1} (1 - F(\hat{L}_{k-1}, \hat{L}_k))\right] = p_0^{n_0},$$

which proves that W has zero mean.

Variance of W From the proof of Theorem 1, we know that

$$\sqrt{N}\left(\prod_{k=1}^{n_0} (1 - F(\hat{L}_{k-1}, \hat{L}_k)) - p_0^{n_0}\right) \xrightarrow[N\to+\infty]{\mathcal{D}} \mathcal{N}(0, \sigma_{n_0}^2),$$

where $\sigma_{n_0}^2 = n_0(1 - p_0) p_0^{2n_0-1}$. So we have

$$\sqrt{N}(1 - F(\hat{L}_{n_0}) - p_0^{n_0}) \xrightarrow[N\to+\infty]{\mathcal{D}} \mathcal{N}(0, \sigma_{n_0}^2),$$

and by symmetry,

$$\sqrt{N}(F(\hat{L}_{n_0}) - F(L_{n_0})) \xrightarrow[N\to+\infty]{\mathcal{D}} \mathcal{N}(0, \sigma_{n_0}^2).$$

It means that

$$\mathrm{Var}(F(\hat{L}_{n_0}) - F(L_{n_0})) = \frac{1}{N}\sigma_{n_0}^2 + o\left(\frac{1}{N}\right),$$

which concludes the proof of Lemma 2. □

References

Arnold, B.C., Balakrishnan, N., Nagaraja, H.N.: A First Course in Order Statistics. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. Wiley, New York (1992)

Au, S.K., Beck, J.L.: Estimation of small failure probabilities in high dimensions by subset simulation. Probab. Eng. Mech. **16**(4), 263–277 (2001)

Au, S.K., Beck, J.L.: Subset simulation and its application to seismic risk based on dynamic analysis. J. Eng. Mech. **129**(8) (2003)

Barg, A., Blakley, G.R., Kabatiansky, G.A.: Digital fingerprinting codes: problem statements, constructions, identification of traitors. IEEE Trans. Signal Process. **51**(4), 960–980 (2003)

Botev, Z.I., Kroese, D.P.: An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. Methodol. Comput. Appl. Probab. **10**(4), 471–505 (2008)

Botev, Z.I., Kroese, D.P.: Efficient Monte Carlo simulation via the generalized splitting method. Stat. Comput. (2011). doi:10.1007/s11222-010-9201-4

Bucklew, J.A.: Introduction to Rare Event Simulation. Springer Series in Statistics. Springer, New York (2004)

Cérou, F., Del Moral, P., Guyader, A.: A non asymptotic variance theorem for unnormalized Feynman-Kac particle models. Ann. Inst. Henri Poincaré, Probab. Stat. **47**(3) (2011, in press)

Cérou, F., Del Moral, P., Le Gland, F., Lezaud, P.: Genetic genealogical models in rare event analysis. ALEA Lat. Am. J. Probab. Math. Stat. **1**, 181–203 (2006)

Cérou, F., Guyader, A.: Adaptive multilevel splitting for rare event analysis. Stoch. Anal. Appl. **25**(2), 417–443 (2007)

Copy Protection Technical Working Group. www.cptwg.org

Del Moral, P.: Feynman-Kac Formulae, Genealogical and Interacting Particle Systems with Applications. Probability and its Applications. Springer, New York (2004)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc., Ser. B, Stat. Methodol. **68**(3), 411–436 (2006)

Del Moral, P., Lezaud, P.: Branching and interacting particle interpretation of rare event probabilities. In: Blom, H., Lygeros, J. (eds.) Stochastic Hybrid Systems: Theory and Safety Critical Applications. Lecture Notes in Control and Information Sciences, vol. 337, pp. 277–323. Springer, Berlin (2006)

Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science. Springer, New York (2001)

Garvels, M.J.J.: The splitting method in rare event simulation. Thesis, University of Twente, Twente, May 2000

Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T.: Multilevel splitting for estimating rare event probabilities. Oper. Res. **47**(4), 585–600 (1999)

Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**(1), 97–109 (1970)

Johansen, A.M., Del Moral, P., Doucet, A.: Sequential Monte Carlo samplers for rare events. In: Proceedings of the 6th International Workshop on Rare Event Simulation, Bamberg, pp. 256–267 (2006)

Kahn, H., Harris, T.E.: Estimation of particle transmission by random sampling. Natl. Bur. Stand., Appl. Math. Ser. **12**, 27–30 (1951)

Lagnoux, A.: Rare event simulation. Probab. Eng. Inf. Sci. **20**(1), 45–66 (2006)

Le Gland, F., Oudjane, N.: A sequential algorithm that keeps the particle system alive. In: Blom, H., Lygeros, J. (eds.) Stochastic Hybrid Systems: Theory and Safety Critical Applications. Lecture Notes in Control and Information Sciences, vol. 337, pp. 351–389. Springer, Berlin (2006)

Merhav, N., Sabbag, E.: Optimal watermarking embedding and detection strategies under limited detection resources. IEEE Trans. Inf. Theory **54**(1), 255–274 (2008)

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)

Rosenbluth, M.N., Rosenbluth, A.W.: Monte Carlo calculation of the average extension of molecular chains. J. Chem. Phys. **23**, 356 (1955)

Rubinstein, R.: The Gibbs cloner for combinatorial optimization, counting and sampling. Methodol. Comput. Appl. Probab. **4**, 491–549 (2008)

Tardos, G.: Optimal probabilistic fingerprint codes. In: Proc. of the 35th Annual ACM Symposium on Theory of Computing, pp. 116–125. ACM, San Diego (2003)

Tierney, L.: Markov chains for exploring posterior distributions. Ann. Stat. **22**(4), 1701–1762 (1994). With discussion and a rejoinder by the author

van der Vaart, A.W.: Asymptotic Statistics. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge (1998)